Object lesson: discovering and learning to recognize objects

Paul Fitzpatrick

AI Lab, MIT, Cambridge, USA
paulfitz@ai.mit.edu

Abstract. Statistical machine learning has revolutionized computer vision. Systems trained on large quantities of empirical data can achieve levels of robustness that far exceed their hand-crafted competitors. But this robustness is in a sense "shallow" since a shift in context to a situation not explored in the training data can completely destroy it. This is not an intrinsic feature of the machine learning approach, but rather of the rigid separation of the powerfully adaptive training phase from the final cast-in-stone system. An alternative this work explores is to build "deep" systems that contain not only the trained-up perceptual modules, but the tools used to train them, and the resources necessary to acquire appropriate training data. Thus, if a situation occurs that was not explored in training, the system can go right ahead and explore it. This is exemplified through an object recognition system that is tightly coupled with an "active segmentation" behavior that can discover the boundaries of objects by making them move.

1 Introduction

The goal of this work is to build a perceptual system for a robot that integrates useful "mature" abilities, such as object localization and recognition, with the deeper developmental machinery required to forge those competences out of raw physical experiences. The motivation for doing so is simple. Training on large corpora of real-world data has proven crucial for creating robust solutions to perceptual problems such as speech recognition and face detection. But the powerful tools used during training of such systems are typically stripped away at deployment. For problems that are more or less stable over time, such as face detection in benign conditions, this is acceptable. But for problems where conditions or requirements can change, then the line between training and deployment cannot reasonably be drawn. The resources used during training should ideally remain available as a support structure surrounding and maintaining the current perceptual competences. There are barriers to doing this. In particular, annotated data is typically needed for training, and this is difficult to acquire online. But that is the challenge this work addresses. It will show that a robotic platform can build up and maintain a quite sophisticated object localization, segmentation, and recognition system, starting from very little.

2 The place of perception in AI and robotics

If the human brain were a car, this message would be overlaid on all our mental reflections: "caution, perceptual judgements may be subtler then they appear". Time and time

again, the difficulty of implementing analogues of human perception has been underestimated by AI researchers. For example, the Summer Vision Project of 1966 at the MIT AI Lab apparently expected to implement figure/ground separation and object recognition on a limited set of objects such as balls and cylinders in the month of July, and then extend that to cigarette packs, batteries, tools and cups in August [19]. That "blind spot" continues to the current day. But there has been progress. Results in neuroscience continue to drive home the sophistication of the perceptual machinery in humans and other animals. Computer vision and speech recognition have become blossoming fields in their own right. Advances in consumer electronics have led to a growing drive towards advanced human/computer interfaces, which bring machine perception to the forefront. What does all this mean for AI, and its traditional focus on representation, search, planning, and plan execution? For devices that need to operate in rich, unconstrained environments, the emphasis on planning may have been premature:

> "I suspect that this field will exist only so long as it is considered acceptable to test these schemes without a realistic perceptual interface. Workers who have confronted perception have found that on the one hand it is a much harder problem than action selection and that on the other hand once it has been squarely faced most of the difficulties of action selection are eliminated because they arise from inadequate perceptual access in the first place." [6]

It is undeniable that planning and search are crucial for applications with complex logistics, such as shipping and chess. But for robotics in particular, simply projecting from the real world onto some form where planning and search can be applied seems to be the key research problem: "This abstraction process is the essence of intelligence and the hard part of the problem being solved" [4]. Early approaches to machine perception in AI focused on building and maintaining detailed, integrated models of the world that were as complete as possible given the sensor data available. This proved extremely difficult, and over time more practical approaches were developed. Here are cartoon-caricatures of some of them:

- **Stay physical:** Stay as close to the raw sensor data as possible. In simple cases, it may be possible to use the world as its own model and avoid the difficulties involved in creating and maintaining a representation of a noisily- and partially-observed world [4]. Tasks such as obstacle avoidance can be achieved reactively, and [9] gives a good example of how a task with temporal structure can be performed by maintaining state in the world and the robot's body rather than within its control system. This work clearly demonstrates that the structure of a task is logically distinct from the structures required to perform it. Activity that is sensitive to some external structure in the world does not imply a control system that directly mirrors that structure in its organization.
- **Stay focused:** Adopt a point of view from which to describe the world that is sufficient for your task and which simplifies the kind of references that need to be made, hopefully to the point where they can be easily and accurately maintained. Good examples include deictic representations like those used in Pengi [7], or Toto's representations of space [16].

- **Stay open:** Use multiple representations, and be flexible about switching between representations as each run into trouble [17]. This idea overlaps with the notion of encoding common sense [15], and using multiple partial theories rather than searching – perhaps vainly – for single unified representations.

While there are some real conflicts in the various approaches that have been adopted, they also have a common thread of pragmatism running through them. Some ask "what is the minimal representation possible", others "what choice of representation will allow me to develop my system most rapidly?" [15]. They are also all steps away from an all-singing, all-dancing monolithic representation of the external world. Perhaps they can be summarized (no doubt kicking and screaming) with the motto "robustness from perspective" – if you look at a problem the right way, it may be relatively easy. This idea was present from the very beginning of AI, with the emphasis on finding the right representations for problems, but it seemed to get lost once division of labor set in and the problems (in some cases) got redefined to match the representations.

There is another approach to robust perception that has developed, and that can perhaps be described as "robustness from experience". Drawing on tools from machine learning, just about any module operating on sensor input can be improved. At a minimum, its performance can be characterized empirically, to determine when it can be relied upon and when it fails, so that its output can be appropriately weighed against other sources. The same process can be applied at finer granularity to any parameters within the module that affect its performance in a traceable way. Taking statistical learning of this kind seriously leads to architectures that seem to contradict the above approaches, in that they derive benefit from representations that are as integrated as possible. For example, when training a speech recognition system, it is useful to be able to combine acoustic, phonological, language models so that optimization occurs over the largest scope possible [18].

The success of statistical, corpus-based methods suggests the following additional organizing principle to the ones already enunciated :-

- **Stay connected:** Maintain a tight empirical connection between parameters in the system and experience in the world. Machine learning techniques start this connection, but are not usually used to maintain it over time. This will require integrating the tools typically used during training with the deployed system itself, and engineering opportunities to replace hand-annotation.

3 Replacing annotation

Suppose there is some property P of the environment whose value the robot cannot determine in general, but which it *can* determine is special situations. Then there is the potential for the robot to collect training data from such special situations, and learn other more robust (and generally applicable) ways to determine the property P. This process will be referred to as "developmental perception" in this paper.

What kind of special situations and properties might fit this specification? One possibility is interaction with a human, where social cues can help the robot evaluate its current situation. This was explored at our laboratory with Kismet, an expressive "infant-

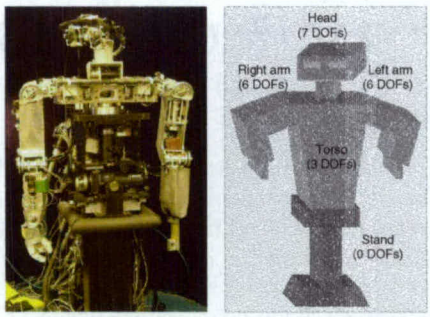

Fig. 1. Degrees of freedom (DOFs) of the robot Cog. The arms terminate either in a primitive "flipper" or a four-fingered hand. The head, torso, and arms together contain 22 degrees of freedom.

like" robot head [2]. Another possibility, and the one this paper explores, is physical exploration, using active perception, on a humanoid robot: Cog.

4 The experimental platform

This work is implemented on the robot Cog, an upper torso humanoid [3]. Cog has two arms, each of which has six degrees of freedom. The joints are driven by series elastic actuators [22]. The arm is not designed to enact trajectories with high fidelity. For that a very stiff arm is preferable. Rather, it is designed to perform well when interacting with a poorly characterized environment, where collisions are frequent and informative events. Cog runs an attentional system consisting of a set of pre-attentive filters sensitive to motion, color, and binocular disparity. The different filters generate information on the likelihood that something interesting is happening in a certain region of the image. A voting mechanism is used to "decide" what to attend and track next. The pre-attentive filters are implemented on a space-variant imaging system, which mimics the distribution of photoreceptors in the human retina as in [20]. The attentional system uses vision and non-visual sensors (e.g. inertial) to generate a range of oculomotor behaviors. Examples are saccades, smooth pursuit, vergence, and the vestibulo-ocular reflex (VOR).

5 Active perception

Active perception refers to the use of motor action to simplify perception [1], and has proven its worth many times in the history of robotics. The most well-known instance of active perception is active vision. The term "active vision" is, in common usage, essentially synonymous with moving cameras. But there is much to be gained by taking advantage of the fact that robots can be actors in their environment, not simply passive observers. They have the opportunity to examine the world using causality, by

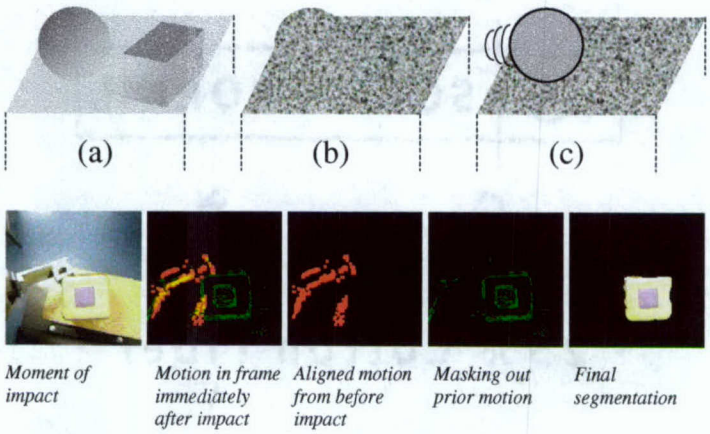

Fig. 2. Cartoon motivation (top) for active segmentation (bottom). Human vision is excellent at figure/ground separation (top left), but machine vision is not (top center). Coherent motion is a powerful cue (top right) and the robot can invoke it by simply reaching out and poking around. The lower row of images show the processing steps involved. The moment of impact between the robot arm and an object, if it occurs, is easily detected – and then the total motion after contact, when compared to the motion before contact and grouped using a minimum cut approach, gives a very good indication of the object boundary [11].

performing probing actions and learning from the response. In conjunction with a developmental framework, this could allow the robot's experience to expand outward from its sensors into its environment.

As a concrete example of this idea, Cog was given a simple "poking" behavior, whereby it selects locations in its environment, and sweeps through them with its arm [10]. If an object is within the area swept, then the motion signature generated by the impact of the arm with that object greatly simplifies segmenting that object from its background, and obtaining a reasonable estimate of its boundary (see Figure 2). The image processing involved relies only on the ability to fixate the robot's gaze in the direction of its arm. This coordination is easy to achieve either as a hard-wired primitive or through learning [10]. Within this context, it is possible to collect excellent views of the objects the robot pokes, and the robot's own arm. This is important, because figure/ground separation is a long-standing problem in computer vision, due to the fundamental ambiguities involved in interpreting the 2D projection of a 3D world. No matter how good a passive system is at segmentation, there will be times when only an active approach will work, since visual appearance can be arbitrarily deceptive.

6 Developmental perception

The previous section showed how, with a particular behavior, the robot could reliably segment objects from the background (even if it is similar in appearance) by poking

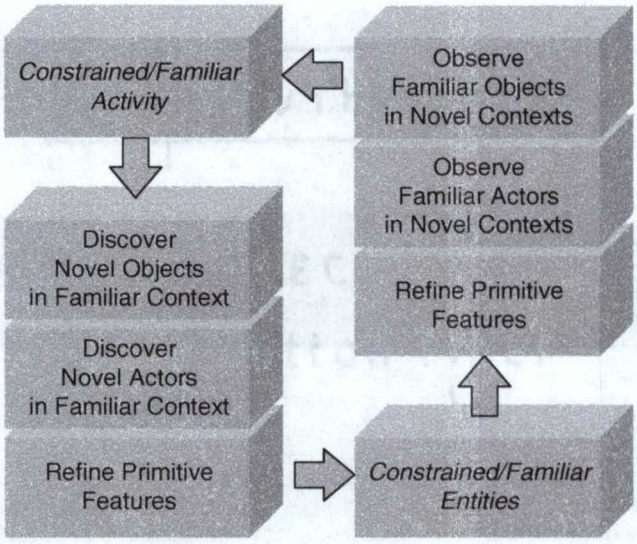

Fig. 3. If the robot is engaged in a known activity (left), there may be sufficient constraint to identify novel elements within that activity. Similarly, if known elements take part in some unfamiliar activity, tracking those can help characterize that activity. Potentially, development is an open-ended loop of such discoveries.

them. It can determine the shape of an object boundary in this special situation, even though it cannot do this normally. This is precisely the kind of situation that a developmental framework could exploit. Figure 3 shows how an open-ended developmental cycle might be possible. Particular, familiar situations allow the robot to perceive something about objects and actors (such as a human or the robot itself) that could not be perceived outside those situations. These objects and actors can be tracked into other, less familiar situations, which can then be characterized and used for further discovery. Throughout, existing perceptual capabilities ("primitive features") can be refined as opportunities arise.

As a specific example of development, the segmented views provided by poking of objects and actors by poking can be collected and clustered as shown in Figure 4. Such views are precisely what is needed to train up an object detection and recognition system, and follow those objects and actors into other, non-poking contexts.

As well as giving information about the appearance of objects, the segmented views of objects can be pooled to train up detectors for more basic visual features – for example, edge orientation.

7 What is orientation?

Natural images are full of anisotropy, where some visual property is very different depending in which direction you measure it. The most obvious example is that of an

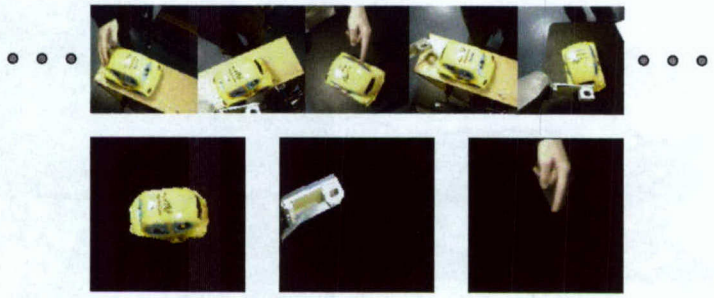

Fig. 4. The top row shows sample views of a toy car that the robot sees during poking. Many such views are collected and segmented as described in [11]. The views are aligned to give an average prototype for the car (and the robot arm and human hand that acts upon it). To give a sense of the quality of the data, the bottom row shows the segmented views that are the best match with these prototypes. The car, the robot arm, and the hand belong to fundamentally different categories. The arm and hand cause movement (are actors), the car suffers movement (is an object), and the arm is under the robot's control (is part of the self).

edge, where there is a discontinuity between one material and another. Visual appearance will typically remain similar in directions parallel to the edge, but could change drastically across the edge. Contours of constant luminance on a shade surface behave like edges also, with luminance change being minimal parallel to the contour and maximal when measured perpendicular to them. For such directional changes in luminance, or any other property, it is natural to associate a direction or *orientation* in which change is minimal. In this chapter, we will be concerned with the orientation associated with edges in luminance at the finest scale available. This is certainly not all that is to be said about orientation (consider, for example, the Kanizsa triangle). But it is a useful case, particularly for object localization and recognition. Orientation detection will prove key to achieving orientation and scale invariance in these tasks.

Orientation is associated with neighborhoods rather than individual points in an image, and so is inherently scale dependent. At very fine scales, relatively few pixels are available from which to judge orientation. Lines and edges at such scales are extremely pixelated and rough. Orientation filters derived from analytic considerations, with parameters chosen assuming smooth, ideal straight lines or edges (for example, [8]) are more suited to larger neighborhoods with more redundant information. For fine scales, an empirical approach seems more promising, particularly given that when the number of pixels involved is low, it is practical to sample the space of all possible appearances of these pixels quite densely. At very fine scales, the interpretation of an image patch could hinge on a relatively small number of pixels. Noise sensitivity becomes a critical issue. But even beyond that, it seems that the assignment of labels to image patches is likely to be quite a non-linear process. One way to avoid making too many assumptions up front is to actually collect some data. Poking allows the robot to build up a

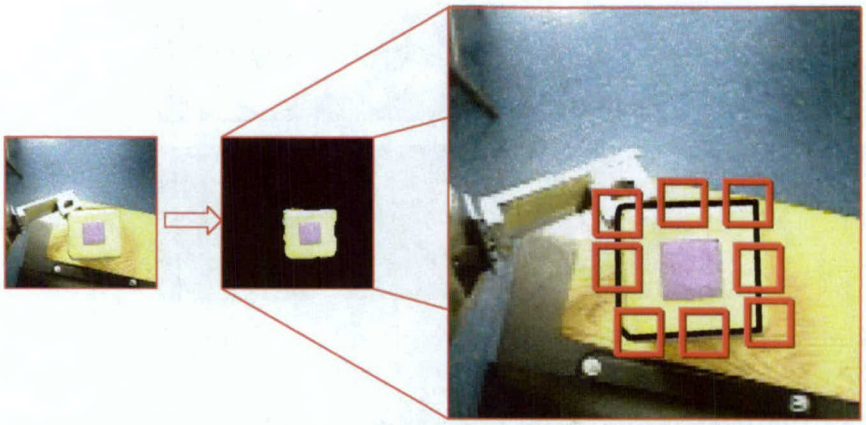

Fig. 5. Sampling an object boundary. Poking (left) identifies the boundary of the object (center); we can then sample along this boundary to get examples of oriented regions of known orientation (right).

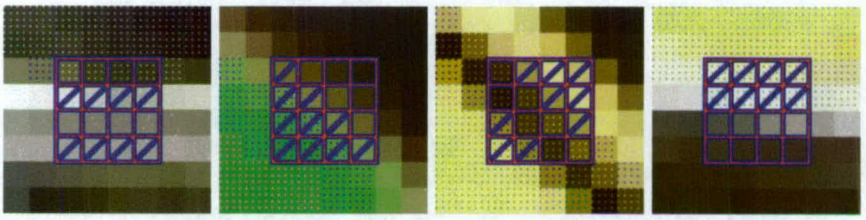

Fig. 6. Some boundary samples. Dotted pixels belong to a segmented object. The four-by-four grid overlaid on the boundary shows the result of thresholding.

reference "catalog" of the manifold appearances real edges can take on. At fine scales, with relatively few pixels, we can hope to explore the space of all possible appearances of such a neighborhood, and collect empirical data on how they relate to orientation. Such statistics are important to minimize the impact of pixel noise, and to capture the scrambling effects of quantization on line and edge appearance.

8 Active segmentation and orientation detection

Once an object boundary is known, the appearance of the edge between the object and the background can be sampled along it, and labelled with the orientation of the boundary in their neighborhood (Figure 5). Figure 7 shows an orientation filter trained up from such data that can work at much finer scales than normally possible when the filter is derived from an ideal edge model such as that of [8]. The "catalog" of edge appearances found shows that the most frequent edge appearances is an "ideal" straight,

Fig. 7. The empirical appearance of edges. Each 4×4 grid represents the possible appearance of an edge, quantized to just two luminance levels. The dark line centered in the grid is the average orientation that patch was observed to have in the training data. The upper set of patches are the most frequent ones that occur in training data consisting of about 500 object segmentations. The lower set of patches are a selection of patterns chosen to illustrated the diversity of possible patterns that can occur. The oriented features represented include edges, thin lines, thick lines, zig-zags, corners etc. It is difficult to imagine a set of conventional filters that could respond correctly to the full range of features seen here – all of which appeared multiple times in object boundaries in real images.

noise-free edge, as might be expected (top of Figure 7) – but a remarkable diversity of other forms also occur which are far less obvious (bottom of Figure 7).

There are powerful object recognition techniques which are robust to occlusion and could be trained with the kind of data poking makes available [21]. This has not yet been done, but it offers the interesting possibility of a computational analog of the kind of development proposed in [13], where segments of a single object visually divided by an occluder become linked perceptually in young infants.

9 Recognition

Using active segmentation, the robot can collect examples of the appearance of objects. Ideally, these examples could be used to train up an object recognition module for those objects, so they could be detected without further physical contact. Object recognition is a vast research area, so it is useful to identify the constraints applicable in this work. The two most important are that real-time performance is required, and that backgrounds and lighting are uncontrolled. For high-speed performance, geometric hashing is a useful technique (for a review see [23]). In this method, geometric invariants of some kind are computed from points in model (training) images, then stored in hash tables. Recognition then simply involves accessing and counting the contents of hash buckets. One possibility for the geometric invariants is to take a set of points selected

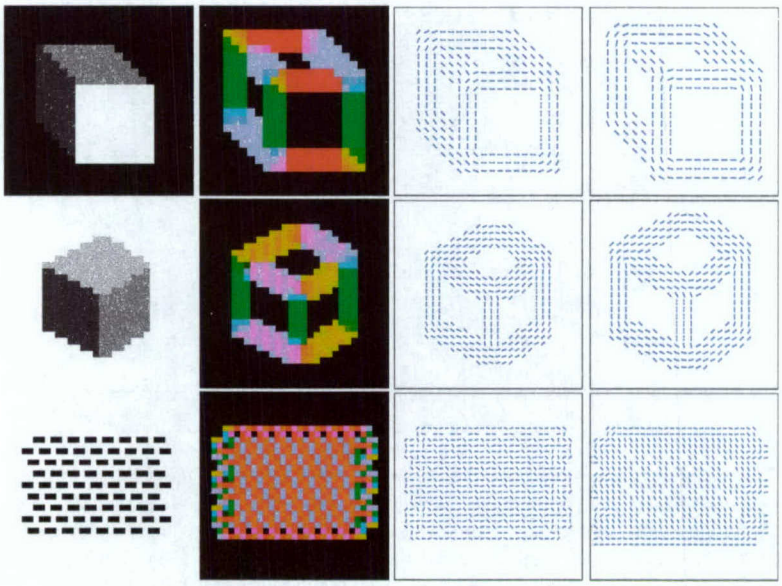

Fig. 8. The orientation filter applied to some test images (on left). Note the small scales involved – the individual pixels are plainly visible. The second column shows the output of the orientation filter, color coded by angle (if viewed in color). The third column shows the same information in vector form. The fourth column shows the orientation determined using steerable quadrature filters [12]. The results are remarkably similar, but the filters are much more computationally expensive to apply. For high resolution patterns, such as that on the bottom row, the empirically-trained orientation filter may have the advantage, but this is difficult to judge.

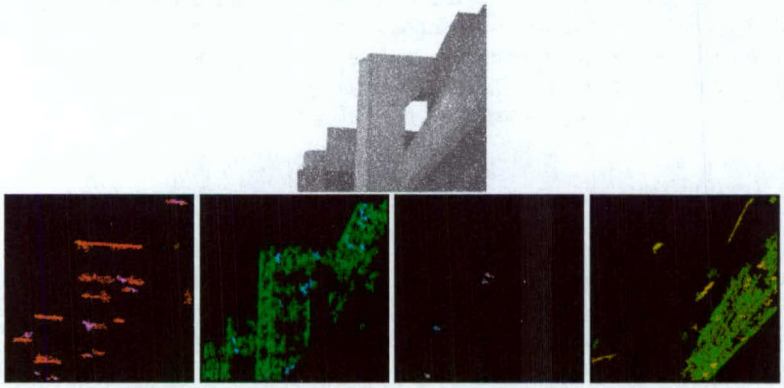

Fig. 9. The orientation filter applied to an image from [5]. The bottom row shows regions with horizontal, vertical, left-diagonal and right-diagonal orientation respectively.

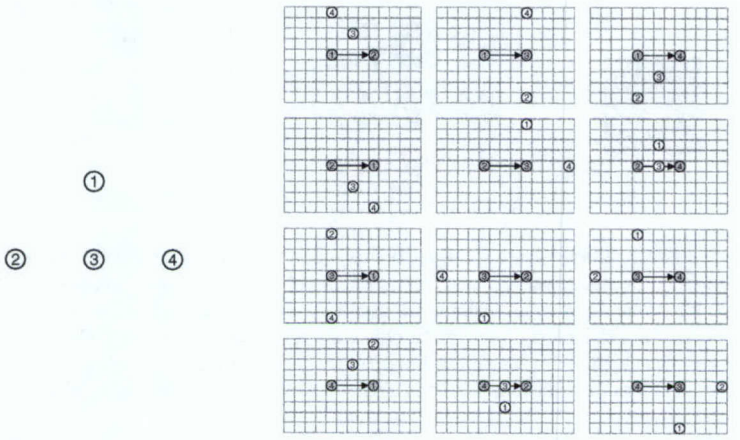

Fig. 10. Geometric hashing for 2D-2D recognition. For the set of points shown on the left, each pair is considered in turn and used to normalize the rest for translation, orientation, and scale. The normalized locations of the points for each permutation are stored in a hash table, along with references to the model and pair of points that generated them.

by an interest operator, and use each pair of points in turn to normalize the remainder by scale and rotation. The position of the normalized points can be stored in a 2D hash table, as shown in Figure 10.

Rather than use pairs of point features, this work uses pairs of oriented regions. The orientation filter developed earlier in this paper is applied to images, and a simple region growing algorithm divides the image into sets of contiguous pixels with coherent orientation. Adaptive thresholding on the minimum size of such regions is applied, so that the number of regions is bounded, independent of scene complexity. In "model" (training) views, every pair of regions is considered exhaustively, and entered into a hash table, indexed by relative angle, relative position, and the color at sample points between the regions. Such features are very selective, yet quite robust to noise.

Another novel feature of the procedure is that rather than simply counting the number of matches with the model, each match is used to predict the center and scale of the object, if the match were valid. Regions with a great deal of convergent evidence are treated as matches. Some variants of geometric hashing do something like this, where each match implies a particular transformation, and only "coherent" matches are aggregated (e.g. [14]), but this method is better suited to objects with symmetries.

Numerical results

Testing on a set of 400 images of four objects (about 100 each) being poked by the robot, with half the images used for training, and half for testing, gives a recognition error rate of about 2%, with a median localization error of 4.2 pixels in a 128×128 image (as determined by comparing with the center of the segmented region given from poking). By segmenting the image by grouping the regions implicated in locating object, and

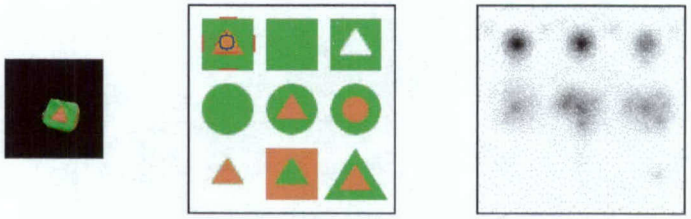

Fig. 11. If the recognition system is trained on real images of a green cube (typical segmentation shown on left), and presented with a synthetic version of the cube along with a set of distractors (middle), we can evaluate what features are used for recognition. The superimposed circle and lines indicate the detected position of the object and the edges implicated. The image on the right shows the strength of evidence for the object across the entire image, which lets us rank the distractors in order of attractiveness. In this case, the most prominent feature used in recognition is the outer green square.

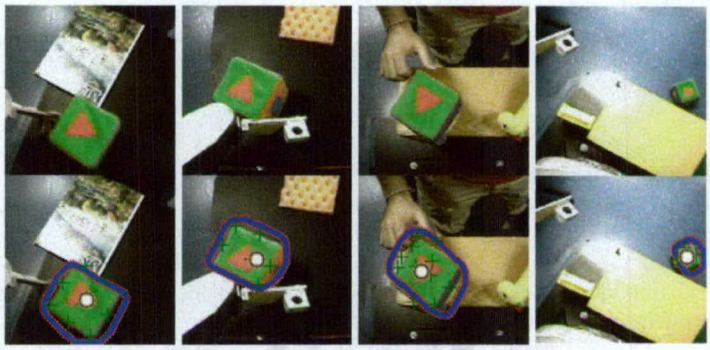

Fig. 12. The cube being recognized, localized, and segmented in real images. The image in the first column is one the system was trained on. The image in the remain columns are test images. Note the scale invariance demonstrated in the final image.

filling in, a median of 83.5% of the object is recovered, and 14.5% of the background is mistakenly included (again, determined by comparison with the results of poking).

Qualitative results

Figure 11 shows a test of the recognition system, trained on real data, and tested on synthetic data. The object in question is a cube with a green face containing a red triangle. When presented with an image containing numerous variations on this theme, the most reasonable match (in the author's judgement) is selected. Figure 12 shows examples of the cube being resegmented in real images. Finally, Figure 13 shows recognition and training occurring online.

10 Conclusions

Robots are very much embedded in a local environment, in a particular context. Rather than trying to use object recognition technology designed to recognize thousands of objects under controlled conditions, either at training time or recognition time, it is more important to look at ways to quickly tailor the robot's perceptual system to its local environment. This paper has given a "proof of concept" for this. Using the simplest possible form of manipulation, simple physical contact, the robot can acquire well-segmented views of an object, and then learn to recognize that object in a variety of situations. If there is some other distractor that the robot confuses the object with, the robot can explicitly learn about that distractor online. While the system presented here is overly simple in some ways – manipulation is very primitive, and the robot relies on human help to bring objects close enough for it to reach – it is robust in a novel and very practical sense: if something isn't working, the robot can take action to fix it.

Acknowledgements

Funds for this project were provided by DARPA as part of the "Natural Tasking of Robots Based on Human Interaction Cues" project under contract number DABT 63-00-C-10102, and by the Nippon Telegraph and Telephone Corporation as part of the NTT/MIT Collaboration Agreement.

Fig. 13. This figure shows stills from a three-minute interaction with Cog. The area of the first frame highlighted with a square shows the state of the robot – the left box gives the view from the robot's camera, the right shows an image it associates with the current view. Initially the robot is not familiar with any objects, so the right box is empty. It is presented with the cube, and pokes it (first frame). Then, if shown the cube again, it recognizes it (this recognition is evidenced by showing an image recorded from when the object was poked). If the cube is turned to another side, the robot no longer recognizes it (third frame). If that side of the cube is presented to the robot to poke (fourth frame), if can then recognize it (fifth frame) and differentiate if from the green side (sixth frame). If it confuses another object with what it has already seen, such as the ball in the seventh frame, this is easily to fix by poking (eighth, ninth frames). The final three frames show the invariance of the recognition system to scale.

References

[1] D. H. Ballard. Animate vision. *Artificial Intelligence*, 48(1):57–86, 1991.

[2] C. Breazeal. *Sociable Machines: Expressive Social Exchange Between Humans and Robots*. PhD thesis, MIT Department of Electrical Engineering and Computer Science, 2000.

[3] R. A. Brooks, C. Breazeal, M. Marjanovic, and B. Scassellati. The Cog project: Building a humanoid robot. *Lecture Notes in Computer Science*, 1562:52–87, 1999.

[4] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence Journal*, 47:139–160, 1991. originally appeared as MIT AI Memo 899 in May 1986.

[5] F. Chabat, G. Z. Yang, and D. M. Hansell. A corner orientation detector. *Image and Vision Computing*, 17(10):761–769, 1999.

[6] D. Chapman. Vision, instruction, and action. Technical report, MIT AI Laboratory, 1990.

[7] David Chapman and Philip E. Agre. Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 268–272, 1987.

[8] J. Chen, Y. Sato, and S. Tamura. Orientation space filtering for multiple orientation line segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):417–429, May 2000.

[9] Jonathan Connell. A colony architecture for an artificial creature. Technical Report AITR-1151, Massachusetts Institute of Technology, 1989.

[10] P. Fitpatrick and G. Metta. Towards manipulation-driven vision. In *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2002.

[11] P. Fitzpatrick. First contact: Segmenting unfamiliar objects by poking them. 2003. submitted to IROS.

[12] T. C. Folsom and R. B. Pinter. Primitive features by steering, quadrature, and scale. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1161–1173, 1998.

[13] S. P. Johnson. Development of object perception. In L. Nadel and R. Goldstone, editors, *Encyclopedia of cognitive science*, volume 3: Psychology, pages 392–399. Macmillan, London, 2002.

[14] B. Lamiroy and P. Gros. Rapid object indexing and recognition using enhanced geometric hashing. In *Proceedings of the 4th European Conference on Computer Vision*, volume 1, pages 59–70, Cambridge, England, April 1996.

[15] Douglas B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[16] Maja J. Mataric. A distributed model for mobile robot environment-learning and navigation. Technical Report AITR-1228, Massachusetts Institute of Technology, 1990.

[17] Marvin Minsky. *The Society of Mind*. Simon and Schuster, New York, 1985.

[18] X. Mou and V. Zue. Sub-lexical modelling using a finite state transducer framework. In *Proceedings of ICASSP'01*, Salt Lake City, Utah, 2001.

[19] S. Papert. The summer vision project. Memo AIM-100, MIT AI Lab, July 1966.

[20] G. Sandini and V. Tagliasco. An anthropomorphic retina-like for scene analysis. *Computer Vision, Graphics and Image Processing*, 14(3):365–372, 1980.

[21] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, January 2000.

[22] M. Williamson. *Robot Arm Control Exploiting Natural Dynamics*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1999.

[23] H. Wolfson and I. Rigoutsos. Geometric hashing: an overview. *IEEE Computational Science and Engineering*, 4:10–21, 1997.